Naval Research Laboratory

Washington, DC 20375-5320

Autonomous Locator of Thermals (ALOFT) Autonomous Soaring Algorithm

Daniel J. Edwards

Offboard Countermeasures Branch
Tactical Electronic Warfare Division

April 3, 2015

REPORT DOCUMENTATION PAGE

1. REPORT DATE (DD-MM-YYYY)
03-04-2015

2. REPORT TYPE
NRL Formal Report

3. DATES COVERED (From - To)

4. TITLE AND SUBTITLE

Autonomous Locator of Thermals (ALOFT) Autonomous Soaring Algorithm

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Daniel J. Edwards

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory, code 5712
4555 Overlook Avenue, SW
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/FR/5712--15-10,272

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
875 N. Randolph Street
Suite 1425
Arlington, VA 22203-1995

10. SPONSOR / MONITOR'S ACRONYM(S)

ONR

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

The Autonomous Locator of Thermals (ALOFT) algorithm goal was to develop an algorithm which could exploit naturally occurring convective thermal updrafts for extending the endurance of an unmanned aerial vehicle (UAV). A commercial off-the-shelf SBXC sailplane was outfitted as a UAV and used for more than 100 test flights of the ALOFT algorithm, with a nominal endurance of 3 minutes after a winch-launch to 100 meters. A notable success was unofficially breaking the cross-country soaring goal-and-return world record by flying 60.4 total miles over approximately 4.55 hours. Best endurance demonstrated by the algorithm was 5.3 hours and best range demonstrated by the algorithm was 70.47 miles. This report documents the final ALOFT software algorithms developed, to capture the most important implementation details and mathematical formulas. Specifically, the soaring algorithm is divided into four major functions: reading state data from the autopilot, identifying the position and other characteristics of a nearby thermal, making soaring behavioral decisions, and sending soaring commands to the autopilot.

15. SUBJECT TERMS

Autonomous soaring	Algorithm	Thermal	Updrafts
Sailplane	UAV	Thermals	Glider

16. SECURITY CLASSIFICATION OF:

a. REPORT	b. ABSTRACT	c. THIS PAGE
Unclassified Unlimited	Unclassified Unlimited	Unclassified Unlimited

17. LIMITATION OF ABSTRACT
Unclassified Unlimited

18. NUMBER OF PAGES
48

19a. NAME OF RESPONSIBLE PERSON
Daniel J. Edwards

19b. TELEPHONE NUMBER (include area code)
202-404-7623

This page
intentionally
left blank

CONTENTS

This page
intentionally
left blank

SYMBOLS

α	Time constant
$\boldsymbol{A}$	Matrix
a	Polynomial term of sink polar
a_n	Polynomial term of sink polar with mass-scaling applied
a_{wind}	Polynomial term of sink polar with wind shifting applied
a_z	Acceleration in z-direction
β	Time constant
B	Y-intercept of a linear interpolation
b	Polynomial term of sink polar
b_n	Polynomial term of sink polar with mass-scaling applied
b_{wind}	Polynomial term of sink polar with wind shifting applied
C	Kalman matrix
C_T	Coefficient of thrust
c	Polynomial term of sink polar
c_n	Polynomial term of sink polar with mass-scaling applied
c_{wind}	Polynomial term of sink polar with wind shifting applied
D	Distance from updraft center, m
d_{prop}	Propeller diameter, m
ϵ	Error
E_{err}	Wind estimator error of east wind speed, m/s
E_{kin}	Kinetic energy
E_{pot}	Potential energy
E_{radius}	Radius of the earth, 6,378,137 m
E_{rot}	Rotational energy
E_s	Total specific energy
E_{total}	Total energy
g	Gravitational constant, 9.81 m/s^2
h	Height, m
I	Moment of inertia
J	Advance ratio
k	Weight scaling factor
K	Kalman matrix
λ	Fit parameters
lat	Latitude
lon	Longitude
m	Mass, kg
M	Slope of a linear interpolation
m_{ac}	Mass of aircraft, kg
m_{polar}	Mass used for polar measurement, kg
n	Load factor
N	Number of elements

Ω_{prop}	Rotation rate of propeller, revolutions per second
N_{err}	Wind estimator error of north wind speed, m/s
P_{atmo}	Power of the atmospheric winds
P_{drag}	Power (loss) due to drag
P_{motor}	Power from the motor
P_s	Specific total power
P_{total}	Total power
P_{wind}	Wind estimator covariance
Q_{wind}	Wind estimator process noise
$queue$	Generic array of data
ρ	Air density, kg/m^3
ρ_∞	Air density at aircraft location, kg/m^3
R	Characteristic updraft radius, m
r^2	Coefficient of determination
R_{wind}	Wind estimator measurement weight
SSE	Sum-squared error
SST	Total sum of squares
τ	Time constant, tau
t	Time, s
$t_{elapsed}$	Elapsed time, s
t_{last}	Previous time step, s
T	Thrust
V_a	Airspeed, m/s
V_{bias}	Airspeed sensor bias, m/s
V_d	Inertial speed down toward the center of the earth, m/s
V_e	Inertial speed to the east, m/s
V_g	Ground speed, m/s
V_h	Headwind speed, m/s
V_{ias}	Indicated airspeed, m/s
V_n	Inertial speed to the north, m/s
V_{sink}	Sink rate, m/s
V_{stf}	Speed to fly velocity, m/s
V_t	True airspeed, m/s
$\hat{V}_t$	Estimated true airspeed, m/s
V_w	Wind velocity, m/s
ω	Rotational rate, deg/s
W	Updraft characteristic strength, m/s
w_{expect}	Expected updraft climb rate, m/s
w_{fit}	Array of fitted updraft strength data
w_{guess}	Array of fitted updraft strength with guessed updraft parameters
w_t	Vertical wind velocity, m/s
W_{hw}	Headwind on aircraft, m/s
W_n	Wind speed from the north, m/s
W_e	Wind speed from the east, m/s
X	Nonlinear transformed distance from updraft center, m
X_{wind}	Wind estimation states
x	Aircraft position, m
x_{drift}	Position due to wind drift, m
x_{orbit}	X-position of orbit center

x_0	Updraft position estimate, m
χ	Ground track angle
y	Aircraft position, m
Y	Nonlinear transformation of updraft strength measurements
y_{drift}	Position due to wind drift, m
y_{orbit}	Y-position of orbit center
y_0	Updraft position estimate, m

This page
intentionally
left blank

EXECUTIVE SUMMARY

The Autonomous Locator of Thermals (ALOFT) algorithm was developed under the Naval Research Laboratory (NRL) Autonomous Soaring 6.2 Base Program from FY2008 to FY2010 in conjunction with North Carolina State University. The goal was to develop an algorithm that could exploit naturally occurring convective thermal updrafts for extending the endurance of an unmanned aerial vehicle (UAV). Essentially, the algorithm senses areas of rising air and commands an autopilot to orbit within them, resulting in the aircraft gaining altitude.

A commercial off-the-shelf RnR Products SBXC sailplane was outfitted as a UAV and performed more than 100 test flights of the ALOFT algorithm, with a nominal endurance of 3 min after a winch-launch to approximately 100 m altitude. A notable success was unofficially breaking the cross-country soaring goal-and-return world record by flying 97.2 km (60.4 mi) declared distance over approximately 4.55 hr. Best endurance demonstrated by the algorithm was 5.3 hr and best range demonstrated by the algorithm was 113.4 km (70.47 mi) open distance. There was no motor on the ALOFT sailplane for any of these flights, so all the endurance and range performance clearly came from flying in thermals.

The same ALOFT algorithm was also used to autonomously soar with Ion Tiger, a battery-electric UAV. Propulsion terms were developed for Ion Tiger and the rest of the ALOFT algorithm was unmodified. These tests showed the applicability of the soaring algorithms to a non-sailplane UAV, wherein the motor power consumption can be reduced or turned off for a time by flying in updrafts.

This report documents the final ALOFT software algorithms in an attempt to capture the most important implementation details and mathematical formulas. Specifically, the soaring algorithm is divided into four major functions: reading state data from the autopilot, identifying the position and other characteristics of a nearby thermal, making soaring behavioral decisions, and sending soaring commands to the autopilot.

This page
intentionally
left blank

AUTONOMOUS LOCATOR OF THERMALS (ALOFT)
AUTONOMOUS SOARING ALGORITHM

INTRODUCTION

The increasing use of unmanned aerial vehicles (UAVs) in military and civilian applications has been accompanied by a growing demand for improved endurance and range. These demands have typically been addressed by improvements in aerodynamic and structural efficiency, improved fuel-efficient propulsion systems, and the ongoing miniaturization of onboard computer and payload systems.

Recently, more attention has been given to the extraction of energy from the atmosphere [1,2,3] as a way to improve performance. Aircraft can make use of atmospheric updrafts, or thermals, to gain altitude without expenditure of onboard fuel stores. By intelligently tracking thermals, an unmanned aircraft can extend its range or endurance without carrying additional fuel or specialized sensors. The trade-off is that the aircraft stops progress along a path, orbits for a duration while gaining altitude, then carries on along the path while gliding with the motor turned off (if equipped with a propulsion system).

The proposed concept of operations (CONOPS) for an autonomous soaring algorithm would be a mission that does not require exact geolocation of the aircraft at all times. Missions that can grant the aircraft an operational volume for maneuvering, such as communications relay missions, are most appropriate for autonomous soaring technologies. It has been proposed, however, that exact geolocation missions, such as tracking a moving vehicle, can be enabled by cooperative autonomous soaring, by allowing multiple aircraft to communicate information about discovered updrafts [4,5]. This report focuses on the core techniques for single-vehicle soaring.

From FY2008 to FY2010, the Naval Research Laboratory (NRL), in conjunction with North Carolina State University, developed the Autonomous Locator of Thermals (ALOFT) autonomous soaring algorithm and demonstrated the effectiveness of using autonomous soaring to increase UAV endurance [2].

Enabling autonomous soaring required the development of an outer control loop to the autopilot subsystem, since most UAV autopilots attempt to reject atmospheric perturbations. ALOFT quantifies and actively utilizes atmospheric perturbations. First, a thermal identification process compares autopilot sensor data to a thermal model, allowing the soaring algorithm to determine when it is flying in a thermal. Next, a soaring manager process chooses the best behavior to both enable energy extraction and balance between soaring and timely progress along the waypoint course. Finally, the determined commands are sent into the autopilot. The ALOFT algorithm was implemented on an RnR Products (Milpitas, CA) SBXC sailplane converted into a UAV for flight-testing, shown in Fig. 1. The sailplane's nominal endurance was 3 minutes after a winch-launch to approximately 100 m altitude.

The ALOFT algorithm was tested in both the eastern and western regions of the United States: in a California high desert and in the inner coastal plain region of North Carolina. ALOFT unofficially broke the goal-and-return unofficial world record by flying 97.2 km (60.4 mi) round-trip in North Carolina in

early October 2008. ALOFT's longest distance flight of 113.4 km (70.47 mi) occurred in California in May 2009. The best endurance demonstrated by the algorithm was 5.3 hours. During 164 total flights and over 70 flight hours on the UAV, approximately 20 hours were spent actively soaring in thermals in autonomous soaring mode.

Atmospheric convection varies with geography, ground cover, weather conditions, and innumerable other factors. Several soaring weather prediction meteorology tools are available for assessing the soaring potential of a specific region [6,7], but none predicts on the micro scale of a single convective updraft. ALOFT carried no weather or mapping information, instead simply running into thermals when flying along a straight path between waypoints. This tactic proved to work sufficiently well for many extended duration flights.

This report documents the ALOFT algorithm for updraft sensing, updraft identification, and soaring guidance, and the mathematical formulas associated with each process.

Fig. 1 — The ALOFT sailplane, shown with the flight crew after a 62-mile cross-country flight

CONTROL LOOP

The ALOFT algorithm is primarily four main functions that run in a loop at 4 Hz, and one function that runs at 20 Hz, shown in Fig. 2. Before the first control loop run, the aircraft physical parameters, thermal/environmental parameters, and Piccolo autopilot [8] connection parameters are defined.

Get State → ID Thermal → Soaring Manager → Send to Piccolo
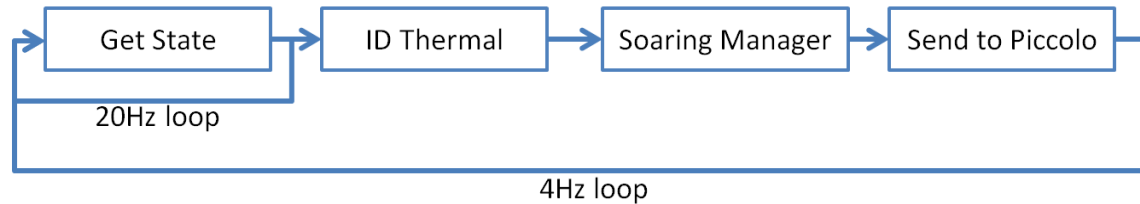
20Hz loop

4Hz loop

Fig. 2 — ALOFT main control loop

The inner Get State loop rate runs faster than the outer loop to smooth out noisy sensor readings. The Piccolo telemetry is available on a COM port at 20 Hz, so 20 Hz was established as the update rate.

The main loop rate is primarily determined by the available computational speed and power consumption, but some physical limitations come into play as well. Main loop rates faster than 4 Hz were tested (up to 10 Hz), but did not appear to have a positive effect on the success of the thermal centering. Slower rates down to 1 Hz were tested, but appeared to consistently fly through lift and have poor centering ability. The rate of 4 Hz appeared to be a good balance between computational usage and thermalling performance.

The soaring algorithm is divided into four major functions: Get State reads the navigation solution from the autopilot; ID Thermal determines the position, strength, and radius of a nearby thermal; Soaring Manager makes behavioral decisions such as orbit location and airspeed to fly; and Send To Piccolo pushes soaring commands to the autopilot.

The main control loop is set up in this modular block process to anticipate future improvements in the individual functions. For example, a better quality state estimation in Get State can be implemented without requiring modification in the ID Thermal block. Similarly, most of the soaring behaviors are controlled in Soaring Manager, whereas the ID Thermal focuses on estimating the environmental parameters of a nearby updraft. In contrast, other autonomous soaring projects [9–12] integrate the guidance algorithm with the atmospheric sensing and forego an independent thermal identification method. While the tightly integrated algorithms are more computationally efficient, the modular nature of the ALOFT approach is more generic and makes testing of individual functions significantly easier.

GET STATE

The Get State function is a combination of requesting information from the autopilot's sensor navigation solution and calculating some additional values based on this data. Specifically, an additional wind estimator and specific power (energy rate) estimator are needed for soaring. The result of Get State is a navigation solution which includes the additional soaring-specific data. Figure 3 shows the process.
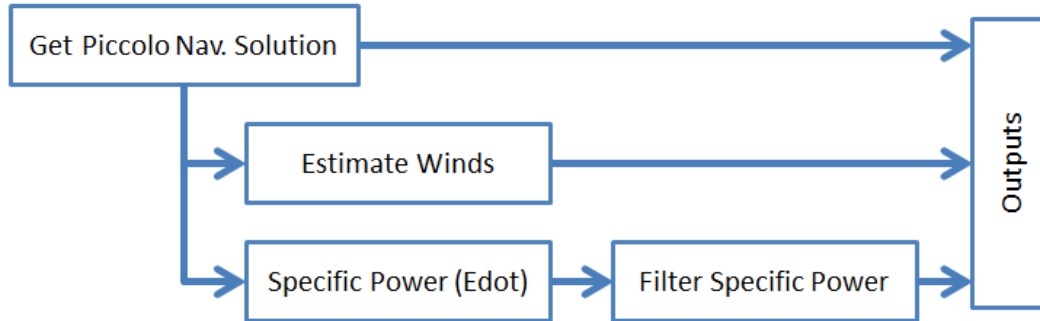
Fig. 3 — Get State function blocks

Get Piccolo Navigation Solution

This block is primarily low-level communication with the autopilot to capture the current aircraft navigation solution. Several items are requested and stored:

- Autopilot time
- Current waypoint number being tracked
- Roll, pitch, and yaw angles
- Roll, pitch, and yaw rates
- GPS north, east, and down velocities
- GPS ground track angle and speed
- Latitude and longitude
- Barometric altitude
- Indicated airspeed
- True airspeed
- West and south wind component estimates
- Propeller rpm

Note that any alternate autopilot that provides this same information, or enough information to derive the rest of the list, can be used instead of the Piccolo.

Estimate Winds

In the author's experience with the Piccolo autopilot system, the Piccolo's wind estimator tends to both overpredict the wind magnitude and to swing the wind direction around as the aircraft heading changes. This is not desired for soaring, as the initial turns into a thermal need to establish the thermal's drift direction within the first one or two orbits, or risk not being able to track it. With a better estimate of the wind components provided in the algorithm below, the vehicle is able to track thermal motion more accurately and find thermals further back in time, greatly improving the usefulness of the algorithm as a whole.

The new method is based on the wind estimator used on the NRL CICADA Mk 3 micro air vehicle [13]. An extended Kalman filter (EKF) was designed to estimate the airspeed sensor bias and the wind

components based on the ground speed, ground track angle, and true airspeed. This method gives much smoother wind data and does not suffer from the same issues as the Piccolo wind estimator.

The Kalman filter states, X_{wind}, to be estimated are true airspeed bias, $\widehat{V}_{bias}$, and wind north and east components, W_n and W_e, respectively:

$$X_{wind} = \begin{bmatrix} \widehat{V}_{bias} \\ \widehat{W}_n \\ \widehat{W}_e \end{bmatrix} \tag{1}$$

On first run, the states are initialized using zeros for each of the states. The initial covariance is

$$P_{wind} = \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{bmatrix} \tag{2}$$

The process noise for the wind is chosen as

$$Q_{wind} = \begin{bmatrix} 0.0001 & 0 & 0 \\ 0 & 0.001 & 0 \\ 0 & 0 & 0.001 \end{bmatrix} \tag{3}$$

Finally, the measurement noise weight is

$$R_{wind} = 0.5 \tag{4}$$

Each run of the filter begins with integrating the covariance matrix:

$$P_{wind}^+ = P_{wind}^- + Q_{wind} \tag{5}$$

Next, the measurement is taken by performing the following:

$$V_n = V_g * \cos(\chi) \tag{6}$$

$$V_e = V_g * \sin(\chi) \tag{7}$$

where V_n and V_e are north and east inertial velocities, V_g is ground speed, and ground track angle is χ. All these are available from GPS data. The errors in wind velocities are specified as

$$N_{err} = V_n - \widehat{W}_n \tag{8}$$

$$E_{err} = V_e - \widehat{W}_e \tag{9}$$

where W_n and W_e are the estimated north and east wind velocities and N_{err} and E_{err} are the wind velocity errors. The airspeed magnitude is computed:

$$|V_a| = \sqrt{N_{err}^2 + E_{err}^2} \tag{10}$$

True airspeed estimate of the sensor reading (which includes bias) is calculated by

$$\tilde{V}_{t_{sensor}} = |V_a| - \hat{V}_{bias} \tag{11}$$

The error to minimize is the difference between the Piccolo's true airspeed reading and the filter's estimate of true airspeed:

$$\epsilon = V_{t_{sensor}} - \tilde{V}_{t_{sensor}} \tag{12}$$

If $|V_a|$ is greater than zero, the standard Kalman filter calculations are performed:

$$C = \begin{bmatrix} -1 & \frac{-N_{err}}{|V_a|} & \frac{-E_{err}}{|V_a|} \end{bmatrix} \tag{13}$$

$$K = P_{wind}C^T(CP_{wind}C^TR)^{-1} \tag{14}$$

Finally, we do the state and covariance updates:

$$X^+_{wind} = X^-_{wind} + K\epsilon \tag{15}$$

$$P^+_{wind} = P^-_{wind} - KCP^-_{wind} \tag{16}$$

The outputs of the wind estimator relating to wind are

$$\widehat{W_n} = X_{wind_2} \tag{17}$$

$$\widehat{W_e} = X_{wind_3} \tag{18}$$

$$|V_w| = \sqrt{\widehat{W_n}^2 + \widehat{W_e}^2} \tag{19}$$

$$\chi_w = atan2(-\widehat{W_e}, -\widehat{W_n}) \tag{20}$$

$$W_{hw} = -\widehat{W_n}\cos(\chi) - \widehat{W_e}\sin(\chi) \tag{21}$$

where χ_w is the wind direction and W_{hw} is the instaneous headwind on the aircraft. The estimated wind north and east values are much cleaner than the Piccolo's estimate and are used throughout the rest of the algorithm, shown for an example snippet of flight data in Fig. 4.

The outputs of the wind estimator relating to airspeed are

$$\hat{V}_{bias} = X_{wind_1} \tag{22}$$

$$\tilde{V}_t = \hat{V}_{bias} + \tilde{V}_{t_{sensor}} \tag{23}$$

The value $\tilde{V}_t$ is a much cleaner version of true airspeed that is used throughout the rest of the ALOFT algorithm in lieu of Piccolo's true airspeed value.
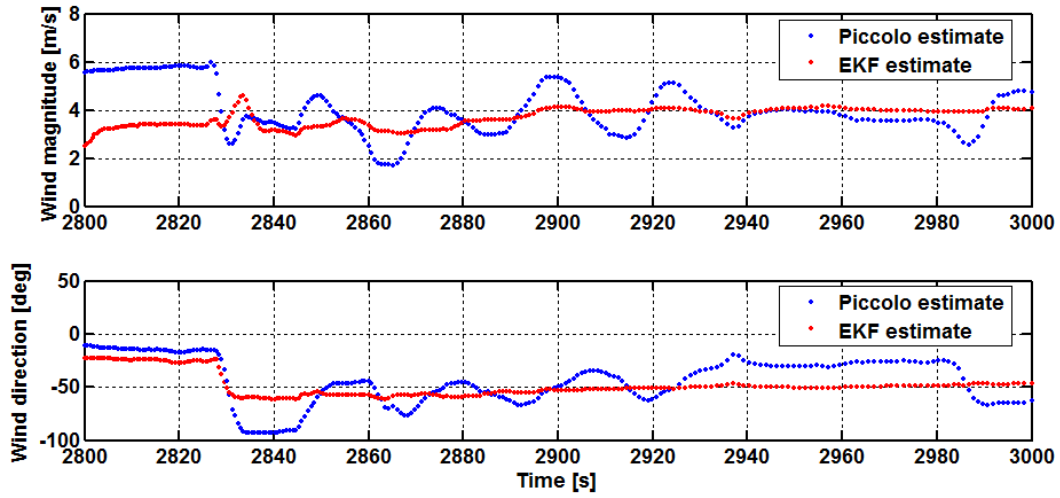
Fig. 4 — Comparison of Piccolo and extended Kalman filter (EKF) wind magnitude and direction estimates

Estimate Specific Energy Rate

The quality of thermal identification, and thus the performance of the energy extraction, depends on a good estimate of the vehicle's energy and power. This estimation includes accounting for any known inputs or losses, such as aerodynamic drag or power input from a propeller. The difference between measured power state and the sources and sinks accounted for directly is assumed to be energy input from the environment. The following method was used for ALOFT, but other estimators might work also. Note that for a sailplane, the power input from the propeller is set to zero.

Power sources and sinks include the following:

$$P_{total} = P_{drag} + P_{motor} + P_{atmo} \qquad (24)$$

Dividing the power terms by weight gives specific power (note the use of weight rather than mass) to get units of velocity:

$$P_s = \frac{P_{total}}{mg} = \frac{P_{drag}}{mg} + \frac{P_{motor}}{mg} + \frac{P_{atmo}}{mg} \qquad (25)$$

The end result is to solve for the specific power of the wind. The total power is determined from a buildup of the known energy sources, potential, kinetic, and rotational:

$$E_{total} = E_{pot} + E_{kin} + E_{rot} \qquad (26)$$

or

$$E_{total} = mgh + 0.5mV^2 + I\omega^2 \qquad (27)$$

Energy stored in rotational rate is small compared to the trades between potential and kinetic energy, and therefore is dropped. Dividing by weight gives specific energy (note the use of weight rather than mass) with units of distance:

$$E_s = \frac{E_{total}}{mg} = h + \frac{V^2}{2g} \tag{28}$$

Differentiating with respect to time provides a rate of change of specific energy:

$$\dot{E}_s = \frac{\dot{E}_{total}}{mg} = \dot{h} + \frac{V\dot{V}}{g} \tag{29}$$

Since power is energy per time, specific energy rate is the same as specific power. Thus, the equivalence holds:

$$\frac{P_{total}}{mg} = \frac{\dot{E}_{total}}{mg} \tag{30}$$

Substituting specific energy rate into the specific power equation and solving for the specific power of the wind term yields

$$\frac{P_{atmo}}{mg} = \dot{h} + \frac{V\dot{V}}{g} + \frac{-P_{drag}}{mg} + \frac{-P_{motor}}{mg} \tag{31}$$

For each location, computing this specific power due to wind provides a measure of the vertical energy available due to atmospheric motion. Conveniently, the units describe the vertical air mass motion as a velocity, coinciding with how an updraft is described physically, as a column of rising air. The following sections describe how each of the terms in this specific power of the wind equation are measured.

Potential Energy

For measuring potential energy, three options are plausible:

- Back-differenced altitude
- GPS vertical velocity
- A pseudo-altitude estimator

Back-differencing altitude was always too noisy for ALOFT, when the altitude data came from the barometric altitude sensor. The pseudo-altitude estimator did not produce good results either, introducing too much lag at any reasonable filtering time constants. Instead, GPS vertical velocity was used because it was relatively smooth and is time-correlated with the measurement. Also, we only need the altitude rate, so absolute accuracy is not necessary. Finally, the specific potential energy rate is

$$\dot{h} = -V_d \tag{32}$$

where V_d is the vertical velocity from the GPS data.

Kinetic Energy

For measuring kinetic energy, three options are plausible:

- Back-differencing airspeed
- Inertial velocity based on acceleration vector
- A pseudo-airspeed estimator

In practice, back-differencing the airspeed data created an extremely noisy signal. The accelerometer-based method worked [14], but tended to show a spike in the signal when rolling into or out of turns. Instead, the pseudo-airspeed estimator was used.

The pseudo-airspeed estimator uses a time constant value, τ, set to 2 in ALOFT. True velocity is used to maintain consistency with altitude and is the input to the estimator. The estimated true velocity, $\hat{V}_t$, comes from the estimated true velocity rate, $\dot{\hat{V}}_t$:

$$\dot{\hat{V}}_t = \frac{V_t - \hat{V}_t}{\tau} \tag{33}$$

The pseudo-acceleration is integrated to get pseudo-velocity:

$$\hat{V}_t = \hat{V}_t + \dot{\hat{V}}_t \Delta t \tag{34}$$

Finally, the specific kinetic energy rate term is calculated using both the pseudo-velocity and pseudo-acceleration states:

$$\frac{V\dot{V}}{g} = \frac{\hat{V}_t \dot{\hat{V}}_t}{g} \tag{35}$$

Drag

It is possible to account for drag in terms of the sink rate of the vehicle in correspondence with its indicated airspeed. This is commonly referred to as a "netto variometer" formulation, since it allows an estimate of the vertical air column directly by removing the aircraft's own contribution to the measured total sink rate. It is also possible to correct the sink polar for load factor, which helps account for the addition of induced drag during steady turns.

First, the aircraft sink polar must be measured accurately. This was accomplished in previous testing [15]. The quadratic polynomial for an SBXC at 5.0 kg is given in Eq. (37) using knots as input and output:

$$V_{sink} = a\,V_t^2 + b\,V_t + c \tag{36}$$

$$V_{sink} = -0.0176V_t^2 + 0.3782V_t - 2.4993 \tag{37}$$

True airspeed is used because it puts the sink rate in units of length, equivalent to the units of altitude; if indicated airspeed was used instead, the sink rate would not correlate correctly to a change in altitude. Next, the polar is shifted based on the actual aircraft mass and the mass correlating with the sink polar polynomials, using scaling factor *k*:

$$k = \sqrt{\frac{m_{ac}}{m_{polar}}} \tag{38}$$

$$V_{sink_{heavy}} = kV_{sink} \tag{39}$$

$$V_{t_{heavy}} = kV_t \tag{40}$$

With these new sink rate and velocity values, new polar polynomial coefficients (a_n, b_n, and c_n) can be determined.

This sink polar is only applicable for an aircraft in level, unaccelerated flight. In a steady turn, the load factor is not unity, but its effect can be accounted for by modifying the sink rate:

$$V_{sink} = V_{sink}n^{1.5} \tag{41}$$

where *n* is the load factor, given by

$$n = \frac{-|a_z|}{g} \tag{42}$$

where a_z is the body vertical acceleration. This correction term helps give more accurate readings when in a steady turn, but there is still some noise when first rolling into an orbit. It is hypothesized this is due to aileron deflection drag and from the inertial measurement unit (IMU) sensor to aircraft center of gravity (CG) offset.

The aircraft sink rate (in units of velocity) is computed using the mass-scaled polynomials (denoted with an *n* subscript), the current aircraft velocity, and the load factor:

$$V_{sink} = (a_nV_t^2 + b_nV_t + c_n)n^{1.5} \tag{43}$$

Finally, the specific power contribution from drag is computed with the following substitution:

$$\frac{P_{atmo}}{mg} = V_{sink} = (a_nV_t^2 + b_nV_t + c_n)n^{1.5} \tag{44}$$

Thrust

For a propeller-powered vehicle, thrust will show up as vertical wind if not accounted for in the power balance.

Initially in the ALOFT implementation on the battery-electric Ion Tiger UAV, the electrical power input to the propulsion system was measured as a means to measure the power term, but this requires knowing the efficiency of several components (motor, speed control, gearbox, propeller) at various environmental conditions to determine the power output to the air.

A much simpler solution for measuring propulsion system power is to fully characterize the propeller over the advance ratio range and map this to the thrust coefficient. This provides surprisingly clean variometer data and is the suggested method for future implementations.

The propeller rpm from the aircraft state data is used directly. This is used to compute the current advance ratio, J. The propeller diameter must be the same as in the table lookup.

$$J = \frac{V_t}{\Omega_{prop} d_{prop}} \tag{45}$$

Next, the advance ratio is used in a table lookup to find the coefficient of thrust, shown in Fig. 5. The data shown is for an Aeronaut (aero-naut Modellbau GmbH & Co., Germany) 16x13 propeller with a custom +4 degree yoke.
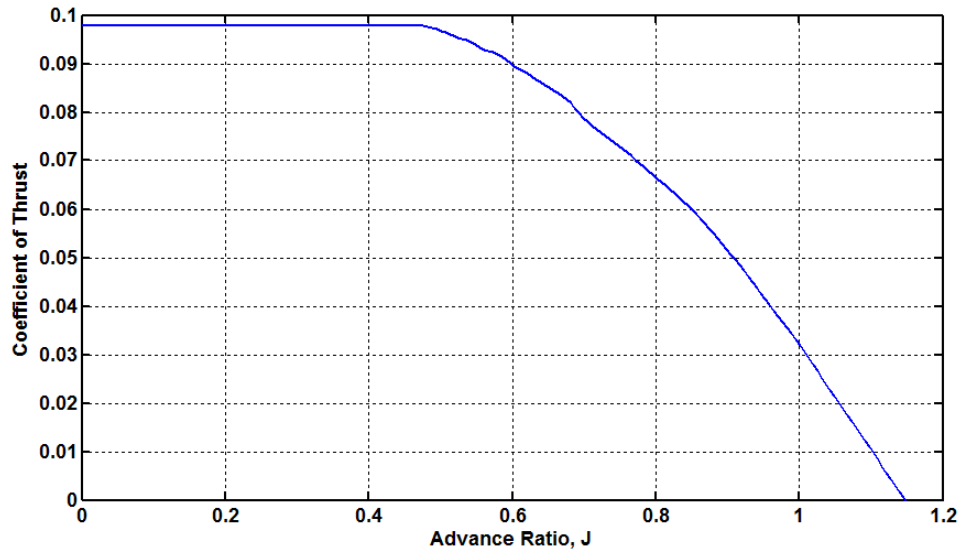

Fig. 5 — Aeronaut 16x13nrl coefficient of thrust curve

Thrust is then calculated using

$$T = \rho_\infty (\Omega_{prop})^2 (d_{prop})^2 C_T \tag{46}$$

Finally, the specific power contribution from the propulsion system is computed:

$$\frac{P_{motor}}{mg} = \frac{V_t T}{mg} \tag{47}$$

Final Combination

The specific energy is calculated and solved for the vertical air mass motion:

$$\frac{P_{atmo}}{mg} = -V_d + \frac{\hat{V}_t \dot{\hat{V}}_t}{g} - (a_n V_t^2 + b_n V_t + c_n)\left(\frac{-a_z}{g}\right)^{1.5} - \frac{\rho_\infty (\Omega_{prop})^2 (d_{prop})^2 C_T}{mg} \tag{48}$$

This equation provides an estimate of the vertical rate of the local air mass taking into account aircraft total energy exchange, aerodynamic sink polar and its changes with aircraft mass, steady-state turns (load factor), and propulsion system effects. Air mass vertical rate is given in units of m/s.

Specific Energy Rate Filter

The Matlab *filtfilt()* function from the Signal Processing Toolbox was used as a smoothing filter on the energy rate queue. This works fabulously to smooth the variometer measurements with zero time-shift, but requires a stored queue. Figure 6 shows the smoothing filter performance on some flight data.
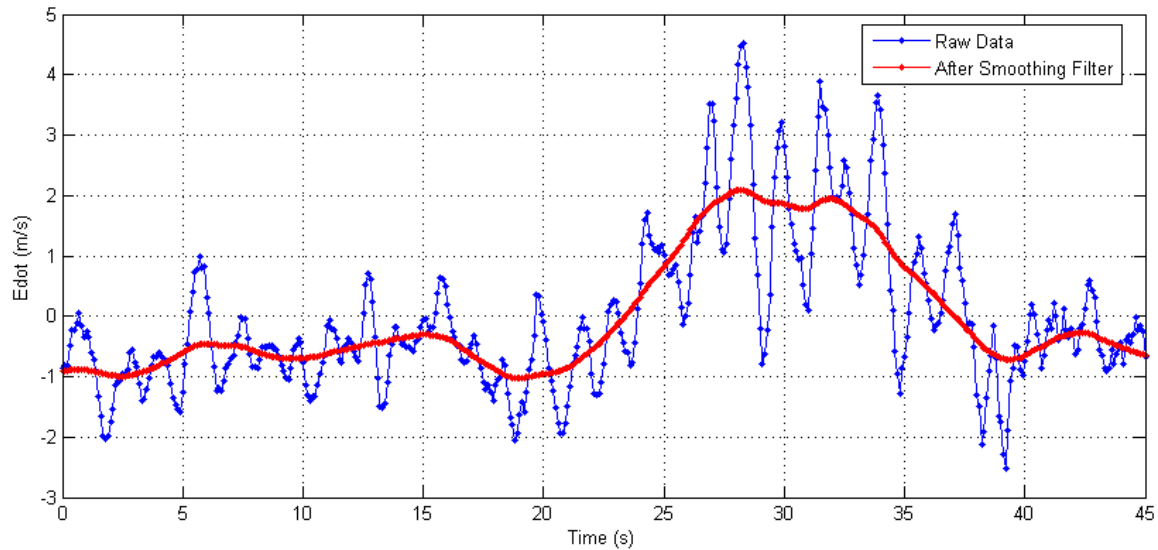

Fig. 6 — Smoothing filter performance on variometer flight data

Alternately, a low-pass filter can be used if a non–batch processing thermal identification method (such as a recursive method) is used, but special attention must be given to maintain time continuity. Another recently explored filter option is a Savitzky-Golay filter suggested by Daugherty and Langelaan [9].

Most important is to have an accurate specific energy rate with no time lag. So any filtering must emphasize no shifting of the energy rate in time. If there is time lag, the vehicle will try to circle in air offset from the actual thermal, resulting in poor thermal centering performance.

THERMAL IDENTIFICATION

Finding the center of a thermal quickly and accurately is of the utmost importance for soaring. A poor soaring controller using better center position data is, in general, better than an expert soaring controller using poor thermal center position data. To quote Welch et al. [16],

> The pilot has three variables to contend with: his angle of bank, his airspeed, and where he flies. The first two are largely interdependent. ... However, both these conditions are insignificant compared to the importance of flying in the right part of the sky.

ALOFT splits the ID Thermal process into its own modular unit so future improvements can be implemented without changing how the controller functions. The thermal identification function has several stages, shown in Fig. 7.
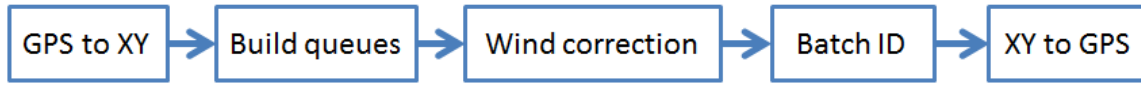

Fig. 7 — ID Thermal process

GPS to XY and XY to GPS

These two functions are coordinate transformations to switch between latitude/longitude and a local, flat-earth coordinate system of distance. This is an approximation which could be done several other ways, including using Universal Transverse Mercator (UTM) coordinates.

Assuming the Earth's radius is 6,378,137 m, converting from GPS to XY is calculated by

$$x = (lat_{home} - lat)E_{radius} \tag{49}$$

$$y = (lon_{home} - lon)E_{radius}\cos(lat) \tag{50}$$

The additional factor in the *y* conversion is an offset for the WGS-84 geoid to account for longitude scaling with latitude, and is only an approximation.

Converting from XY to GPS is calculated by

$$lat = lat_{home} + \frac{x}{E_{radius}} \tag{51}$$

$$lon = lon_{home} + \frac{y}{E_{radius}\cos(lat)} \tag{52}$$

For both conversions, the home position is stored when the soaring program is first started. Approximation errors in the transformation are assumed negligible.

Build Queues

In the batch method of thermal identification, a sliding window of past data is used. Only data stored in the queues is remembered; after the time length of the queue elapses, that data is completely forgotten. Queue lengths of 90, 60, 45, and 25 seconds were all tried on ALOFT with varying levels of success. The final selection of 45 seconds is a reasonable compromise of memory and computational expense.

The Build Queues function handles the creation and updating of the arrays/queues. These data histories are persistent between visits to the function. Several arrays are built and updated each cycle:

- Autopilot time
- X and Y vehicle positions
- Specific energy rate
- West and east wind components

First, all these queues start out empty. With each subsequent visit to the Build Queues function, the current state data is added to each queue in a first-in-first-out (FIFO) manner. In the ALOFT code, the least recent data is stored in the first array index, whereas the most recent data is stored at the end of the array.

In Matlab code, each queue is updated by dropping the first array index and adding the new data to the last array index:

$$queue = [queue(2:end) \quad new_{item}]$$ (53)

Each queue is cut down to ensure that data is only stored for the correct window of queue time length, t_{max}. This starts with shifting the time queue such that it is elapsed time with the end of the array always set to zero. This is primarily for convenience. The last index array should therefore always be zero and the first index array should be ~45 seconds once the queue is full.

$$\bar{t}_{elapsed} = \bar{t} - \bar{t}_{end}$$ (54)

The function *find()* returns the indices of the array that satisfy the test condition

$$index = find\left(\bar{t}_{max} < abs(\bar{t}_{elapsed})\right)$$ (55)

All the queues are cut to length by removing any array indices in *index*. The following Matlab lines complete this action:

$$queue[index] = [\ ]$$ (56)

Each queue is updated using this process with each revisit to the Build Queues block. Thus, the queues stay time-correlated with the time queue, and all of equal lengths.

In this report, queues are equivalent to an array, and are denoted with a bar.

Wind Correction

The wind correction of the position queue is a very important step in understanding how a thermal moves in time. Effectively, thermals move in a drifting coordinate frame (rather than a GPS frame), primarily following the wind direction, but not always. Some thermals captured by ALOFT have appeared to drift upwind, making the wind correction process a target for future improvements.

The ALOFT algorithm uses the wind speed estimate to push the position queue into a drifted frame of reference. First, the winds are averaged over the queue:

$$x_{drift} = mean(\overline{W_n}) \tag{57}$$

$$y_{drift} = mean(\overline{W_e}) \tag{58}$$

Then, the drift correction is applied to each of the position queue's items using the elapsed time to account for any data dropouts:

$$\bar{x}_{drifted} = \bar{x} + \bar{t}_{elapsed} x_{drift} \tag{59}$$

$$\bar{y}_{drifted} = \bar{y} + \bar{t}_{elapsed} y_{drift} \tag{60}$$

Batch ID

The batch thermal identification method has proven rather robust and flew on most ALOFT test flights. The Batch ID process relies on the curve fit coefficient of determination, described below in the section on regression analysis, as the optimization parameter. In general, Batch ID consists of testing 34 different locations as candidate thermal centers and selecting the location with the highest value of the curve fit confidence. Figure 8 shows the general process.
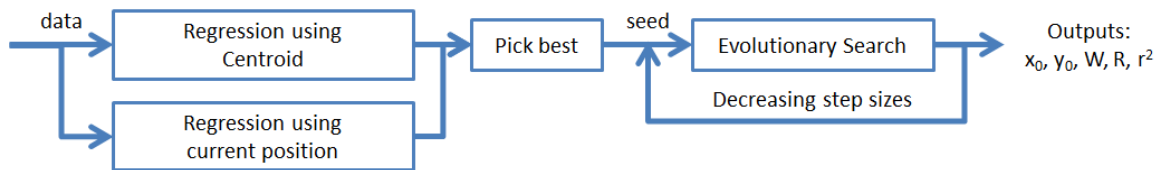

Fig. 8 — Batch ID process

First, Allen's centroid method [1] determines a candidate thermal position. This position is run through the regression function and the fit confidence value stored. Next, the aircraft's current position is used as a candidate position. This position is run through the regression function and the fit confidence value stored. These two fit confidence values are compared, and the position with the higher confidence is chosen as the seed for the next series of steps.

It is worth noting that Allen's centroid method can predict a center position only within the extent of the position measurements since it is analogous to a center-of-mass formulation. This means the centroid method tends to predict a thermal center position well behind the aircraft in time, particularly becoming an issue if the aircraft is flying in a straight line and has not yet reached or has not yet passed a thermal. Assessing the fit confidence of the aircraft's current location allows testing a point closer to the actual thermal, addressing this deficiency of the centroid method.

Next, an evolutionary search is run with the determined initial seed position. Using the seed as the center position, eight new positions spaced in a circle around the center are all tested for their fit confidence. The highest confidence position is chosen as the seed for the next search. Five search cycles of this process use decreasing step sizes of 50 m, 35 m, 20 m, and 15 m to determine the highest confidence position. These step sizes were determined experimentally.

After the evolutionary search, the determined position is checked against a user-defined maximum distance from the aircraft (350 m). If the determined thermal position is greater than the maximum allowable, the centroid position is used instead.

Finally, the outputs of the Batch ID function are the thermal position (x_0, y_0), the estimated thermal strength (W), the estimated thermal radius (R), and the fit confidence (r^2).

An alternate unscented Kalman filter (UKF) thermal identification method was tested at Cal Valley, California, in 2009, but was not as accurate as the Batch ID method according to Hazard [14], at the expense of computational cycles. However, the UKF is more computationally efficient.

Centroid Method

The Allen centroid method [1] for finding the center of a thermal is analogous to mass moment of inertia, substituting specific energy rate for mass:

$$x_0 = \frac{\sum(\bar{x}\bar{w}_t^2)}{\sum(\bar{w}_t)} \tag{61}$$

$$y_0 = \frac{\sum(\bar{y}\bar{w}_t^2)}{\sum(\bar{w}_t)} \tag{62}$$

This method has a tendency to predict a thermal center backwards along the thermal path somewhere, even if the measurements show the thermal will be ahead of the flight path. For this reason, it is not a good method for finding the thermal center. It does, however, guarantee to always give a value contained within bounds of the flight path, so it is a reasonable seed position for other methods.

Evolutionary Search

The evolutionary search used in Batch ID in effect is a simple grid search around a center node using a fixed step size. Given a starting center position, each of the nodes is run through the regression analysis and a fit confidence value determined. The node location with the highest fit confidence is chosen as the winner and its position and fit confidence passed back up to the calling function. The grid search is shown in Fig. 9.
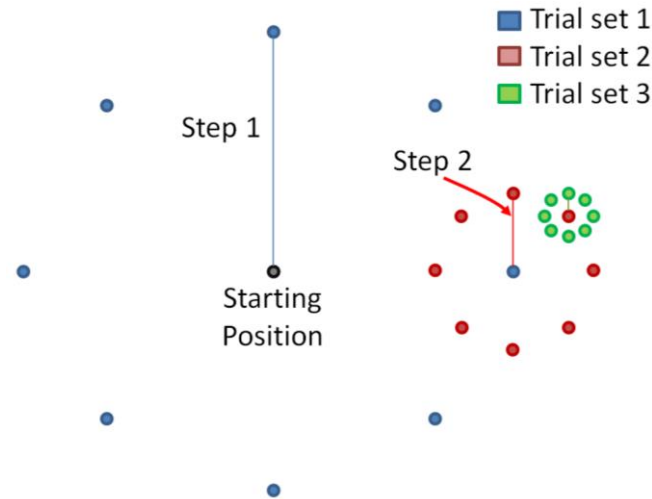
Fig. 9 — Evolutionary search node generation process

Regression Analysis

The regression analysis function encapsulates the nonlinear regression process (Fig. 10). It first starts with nonlinear least squares (NLLS) to initialize the parameters for a 2D nonlinear regression. The output of this analysis for a given position and specific energy rate queue is the following:

- Estimated thermal strength, W
- Estimated thermal radius, R
- Fit confidence, r^2

Notably, both the NLLS and the 2D nonlinear regression use an analytic model of an updraft, based on a Gaussian distribution cross section. All data is assumed at the same altitude, but the position of the data samples are from the drift-corrected position queues.
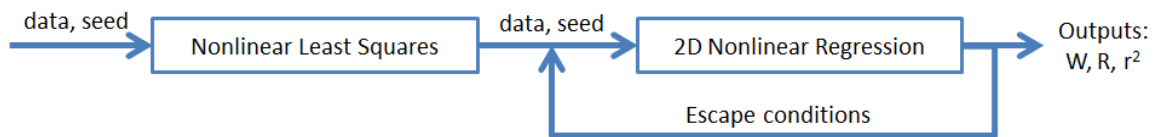

Fig. 10 — Regression analysis loop major steps

Nonlinear Least Squares

The nonlinear least squares method relies on a coordinate transformation to turn the 3D curve fit into a 2D curve fit problem. Specifically, the updraft model is a Gaussian distribution [17] given by

$$\bar{w}_t = W e^{-\left(\frac{\bar{D}}{R}\right)^2} \tag{63}$$

$$D = \sqrt{(\bar{x} - x_0)^2 + (\bar{y} - y_0)^2} \tag{64}$$

where W is the characteristic thermal strength, R is the characteristic thermal radius, w_t is the queue of specific energy rate measurements, x and y are the queue of position measurements, and x_0 and y_0 are the thermal positions.

The nonlinear transformation that makes this equation solvable using linear methods is

$$Y = log\left(w_{t_i}\right) \tag{65}$$

$$X = (x_i - x_0)^2 + (y_i - y_0)^2 \tag{66}$$

With this transformation, standard linear regression is performed on x and y, where N is the number of elements in the array, M is the determined slope, and B is the determined y-intercept [18]:

$$M = \frac{N\sum(\bar{X}\bar{Y}) - \sum(\bar{X})\sum(\bar{Y})}{N\sum(\bar{X}^2) - \left(\sum(\bar{X})\right)^2} \tag{67}$$

$$B = \frac{\sum(\bar{Y}) - M * \sum(\bar{X})}{N} \tag{68}$$

Next, the inverse transformation puts the determined slope and intercept back into the original coordinate system:

$$W = e^B \tag{69}$$

$$R = \sqrt{\frac{-1}{M}} \tag{70}$$

The slope can be positive in certain cases, which makes the estimated R imaginary, or zero, which gives a divide-by-zero error. In either of these cases, ALOFT simply sets the R to be the mean of D.

The coefficient of determination is the fit confidence value. This is computed by using the fitted parameters to calculate the model function:

$$w_{fit_i} = W e^{-\left(\frac{(x_i - x_0)^2 + (y_i - y_0)^2}{R^2}\right)} \tag{71}$$

The sum-squared error is computed using the difference between the model and the measurements:

$$SSE = \sum_i^N \left(w_{fit_i} - w_{t_i}\right)^2 \tag{72}$$

Next, the total sum of squares uses the difference between the model and the mean of the measurement:

$$SST = \sum_i^N \left(w_{fit_i} - \bar{w}_t\right)^2 \tag{73}$$

The coefficient of determination, r^2, is calculated in the non-transformed space by

$$r^2 = 1 - \frac{SSE}{SST} \tag{74}$$

Finally, the determined updraft strength, radius, and coefficient of determination are handed back to the calling function.

2D Nonlinear Regression

This nonlinear regression is the heart and soul of the ALOFT algorithm. Like NLLS, it performs the regression in 2D space. Passed into this function are the following:

- Drift-corrected x- and y-position queues
- Specific energy rate queue
- Thermal position coordinates, x_0 and y_0
- Updraft parameter seeds, W and R

The 2D nonlinear regression holds fixed the starting position coordinates (x_0, y_0) while varying the two updraft parameters (W, R) to find better values. Since the regression is done using a nonlinear model directly, rather than a transformed one, the coefficient of determination calculated here should always be equal to or better than that from NLLS. Outputs of this function are the following:

- Estimated thermal strength, W
- Estimated characteristic thermal radius, R
- Coefficient of determination, r^2

First, the 2D nonlinear regression is seeded with the results from the NLLS method. This is generally sufficient to start in the neighborhood of the correct minima. The initial guess is written as

$$\bar{\lambda} = \begin{bmatrix} W \\ R \end{bmatrix} \tag{75}$$

Then, the following procedure is iterated for a maximum of 10 cycles. The function is evaluated with the estimated parameters:

$$D_i = \sqrt{(x_i - x_0)^2 + (y_i - y_0)^2} \tag{76}$$

$$w_{guess_i} = W e^{-\left(\frac{D_i}{R}\right)^2} \tag{77}$$

The Jacobian of the update Gaussian model function is taken and is defined as matrix $\boldsymbol{A}$:

$$\boldsymbol{A} = \begin{bmatrix} \frac{\partial w_t}{\partial W} & \frac{\partial w_t}{\partial R} \end{bmatrix}^T \tag{78}$$

The Jacobian is created from the measurements, and computed:

$$A = diag\left(\left[e^{-\left(\frac{D}{R}\right)^2}\right]\right) \begin{bmatrix} 1 & \frac{2WD_1^2}{R^3} \\ \vdots & \vdots \\ 1 & \frac{2WD_N^2}{R^3} \end{bmatrix} \tag{79}$$

The difference of the model to the guess is then taken:

$$\bar{B} = \bar{w}_t - \bar{w}_{guess} \tag{80}$$

The pseudo-inverse determines the best fit parameters that minimize the error $\bar{B}$. Note that the inverse must be well conditioned.

$$\overline{\Delta\lambda} = (A^T A)^{-1} A^T \bar{B} \tag{81}$$

The update is

$$\bar{\lambda} = \bar{\lambda} + \overline{\Delta\lambda} \tag{82}$$

For the escape condition, the sum-squared error and the change in sum-squared error are computed:

$$SSE = \bar{B}^T \bar{B} \tag{83}$$

The escape conditions from the iteration are when either of the following is satisfied:

$$|SSE| < 1 \tag{84}$$

$$|\Delta SSE| < 0.01 \tag{85}$$

Once out of the loop, the coefficient of determination is then computed [17]. This starts with the sum-squared error previously computed and the total sum of squares:

$$SST = \sum_i^N \left(w_{t_i} - \bar{w}_t\right)^2 \tag{86}$$

The coefficient of determination is

$$r^2 = 1 - \frac{SSE}{SST} \tag{87}$$

The final estimated outputs are

$$W = \bar{\lambda}_1 \tag{88}$$

$$R = \bar{\lambda}_2 \tag{89}$$

Finally, the final estimated strength (W), characteristic radius (R), and coefficient of determination (r^2) are given back to the calling function. The updraft position (x_0, y_0) is unchanged.

SOARING MANAGER

After the ID Thermal process concludes, the Soaring Manager controls the behavior of what the aircraft does with the information. The best thermal-finding algorithm in the world would be useless without good decisions. Specifically, Soaring Manager determines waypoint and airspeed commands based on telemetry data, user commands, and thermal identification (see Fig. 11).
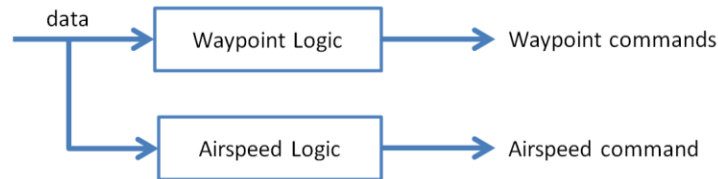

Fig. 11 — Soaring Manager process

The logic that follows is a result of more than 70 hours of autonomous soaring flight and implementing behavioral suggestions from several experienced radio-control pilots.

Soaring Mode Latching/Unlatching

To get the correct behaviors to activate at the right time, a state machine runs through a series of options. The first is the latch engage process, shown in Fig. 12.
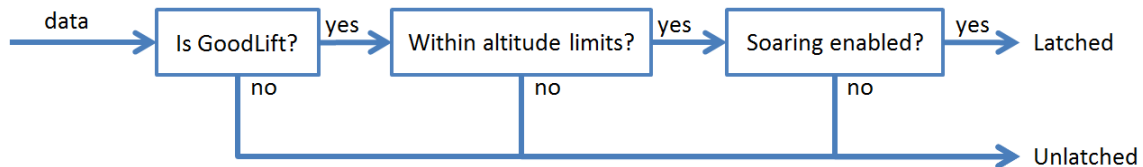

Fig. 12 — Latch engage process

The "Is Goodlift" block examines the output of the ID Thermal and determines if a nearby thermal is good enough quality for soaring, discussed in greater detail in a section below. The "altitude limits" block is a safety process that prevents the aircraft from entering soaring mode if it is too high or too low. This altitude band for ALOFT flying was typically a 500 ft minimum and 5000 ft maximum, giving the safety pilot sufficient altitude to manually set up a safe off-field landing and to prevent the vehicle from climbing beyond visual-line-of-sight limits. The "soaring enabled" block is a user-selectable mode from the ground station. Finally, the output is the soaring mode, either latched or unlatched.

Soaring Mode Actions

The next set of decisions looks at the current soaring mode and makes decisions based on if the mode just changed, shown in Fig. 13.
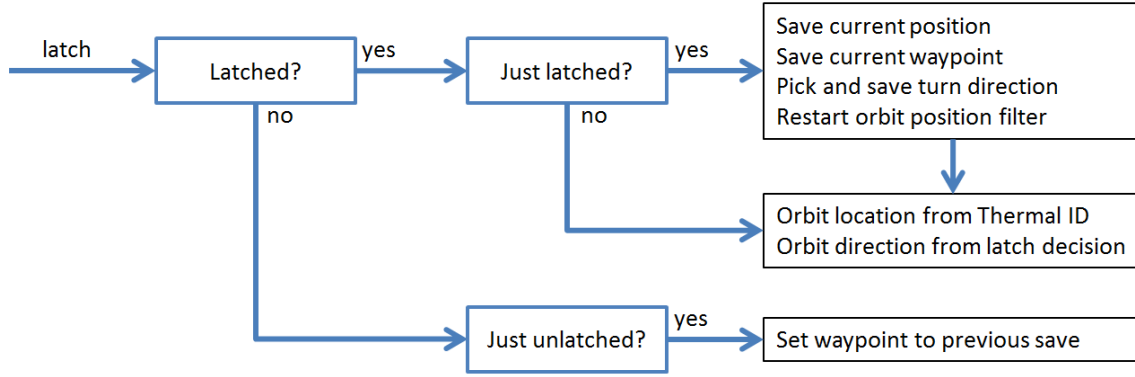

Fig. 13 — Soaring mode actions via latch mode

When the aircraft first switches into latched soaring mode, a number of initialization items occur, such as saving the current position, saving the current waypoint, selecting the turn direction, and restarting an orbit position filter.

Normal latched mode tracks the updraft location determined by ID Thermal and orbits in the direction from the decision made when just latched. Finally, the position is filtered to slow the movement of the orbit location, which tends to jump around somewhat when in turbulent air.

When the aircraft switches into unlatched mode, the aircraft is simply sent to the previous waypoint it was tracking prior to latching.

The orbit position is modified by passing the position through low-pass filter to smooth some position noise from the fitting algorithm:

$$\Delta t = t - t_{last} \tag{90}$$

$$\alpha = \frac{\beta}{\beta - \Delta t} \tag{91}$$

$$x_{orbit} = (\alpha)x_{0_{last}} + (1 - \alpha)x_0 \tag{92}$$

$$y_{orbit} = (\alpha)y_{0_{last}} + (1 - \alpha)y_0 \tag{93}$$

For ALOFT, β is set to 10.

Is GoodLift Determination

The function that determines if the measurements are good enough for the aircraft to begin soaring is a critical function that controls the behavior of how the aircraft looks for thermals. ALOFT's determination is built on the idea that the time history of energy rate is the only information that the algorithm has to determine if the vehicle should latch into soaring or not. Allen used the first and second derivative of specific energy [1], but this method tends to latch early when soaring in turbulent thermals due to the high amount of noise in the second derivative of specific energy. Figure 14 shows the ALOFT GoodLift determination process.
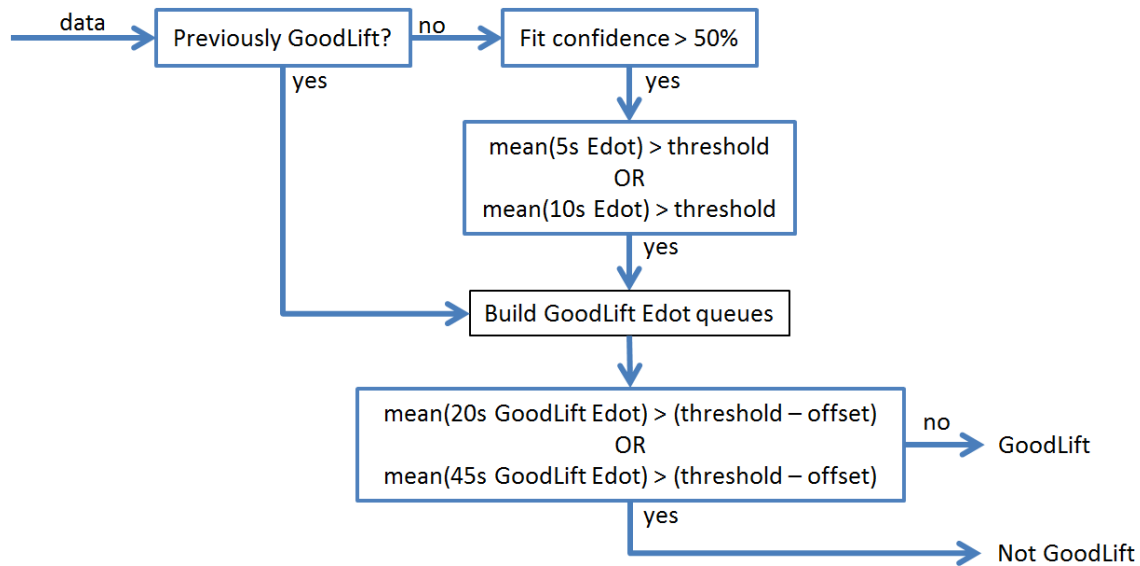

Fig. 14 — GoodLift mode determination

First are the engage methods. If the previous mode was GoodLift, the current mode will still be GoodLift; this essentially locks the mode until a disengage occurs. The thermal identification confidence must be above 50% for the mode to be switched to GoodLift; this ensures that GoodLift is only engaged for reasonably organized thermals. The average energy rate for the previous 5 seconds or 10 seconds must be above a threshold for the mode to be switched to GoodLift. At this point, when the air is considered good for soaring, a FIFO queue accumulates specific energy rate data up to 45 seconds in the past.

Next are the disengage methods. The average of the previous 20 and 45 seconds of the accumulated GoodLift specific energy rate queues is taken and compared to a threshold value to determine if a disengage of GoodLift should occur. An offset to the threshold of 0.5 m/s is allowed to give the disengage process some hysteresis.

A notable behavior falls out of this process: the minimum time in GoodLift mode is 20 seconds. This is important because it gives just enough time for the SBXC to complete approximately one orbit. This single turn is effectively an exploration mode.

For both engaging and disengaging GoodLift, the threshold value is chosen using the altitude-based speed-ring curve discussed in the Speed Ring Settings section. This effectively allows the speed-ring curve to act as a behavior tuning knob.

Turn Direction

One common question in the sailplane community is how to determine the correct direction to turn when latching into a thermal. Generally, an aircraft will not enter a thermal directly at its center, so a determination to turn left or right toward the best part of the thermal has both a correct and an incorrect answer. For ALOFT, this is complicated by the fact that flying in a straight line for the entire queue length of time makes the determination of the thermal being on the left or right side mathematically unobservable. However, ALOFT testing found the following:

- It does not matter which way the aircraft turns initially because the observability increases immediately upon turning, giving the algorithm a good lock on the updraft center. Since the aircraft also knows how the thermal is drifting, it can re-center as part of the entry maneuver. In other words, the ALOFT algorithm is better at centering in the core of a thermal than manually piloted remote-controlled sailplanes, and thus does not rely as heavily on choosing the "correct" turn direction.

- Whichever direction the algorithm chooses initially, it maintains turning that direction. For example, if the aircraft decides to make an initial right-hand turn, the rest of the orbits in this thermal will continue to be to the right. Changing the turn direction means significant drag and a departure from the core of the lift, so reversing the orbiting direction is ill-advised.

- The algorithm has a 50/50 chance of choosing the "correct" turn direction, so any direction guesses only need to be correct more than 50% of the time to improve the odds.

To make the turn direction determination, several possibilities exist:

- Always pick the same direction — In early testing, ALOFT always turned to the right when it latched into a thermal. This worked just fine.

- Look at the aileron deflection at the moment of latching and turn into the up aileron — This method turns toward whichever wing is being pushed upward. The author explored this idea working with data from Allen's CloudSwift and exceeded 85% success rate based on a qualitative assessment of the thermal's location after the vehicle eventually centered.

- Turn the direction the path is already going — ALOFT currently looks at the previous four points in the position queue, draws a line through them, determines if the thermal position estimate is to the left or right of that line, then turns toward the thermal.

The turn direction algorithm consists of the following:

- Cut position queues down to the most recent four data points.
- Run a linear regression through these points.
- Determine the aircraft's heading from this linear regression.
- Determine the identified thermal position's heading.
- If the thermal is to the left of the aircraft's heading, the turn direction decision is left.
- If the thermal is to the right of the aircraft's heading, the turn direction decision is right.

This direction decision function is run each time through ID Thermal, but only matters immediately on a latch decision, when the turn direction is locked in (until the next latch decision).

Airspeed State Machine

Airspeed command is handled independently of the waypoint path, primarily based on altitude, shown in Fig. 15. The additional speed option above 1.3 times maximum altitude is to ensure the aircraft can get out of any aggressive lift; this option was added after an experience in which ALOFT continued to climb after unlatching from a very large thermal at maximum altitude.
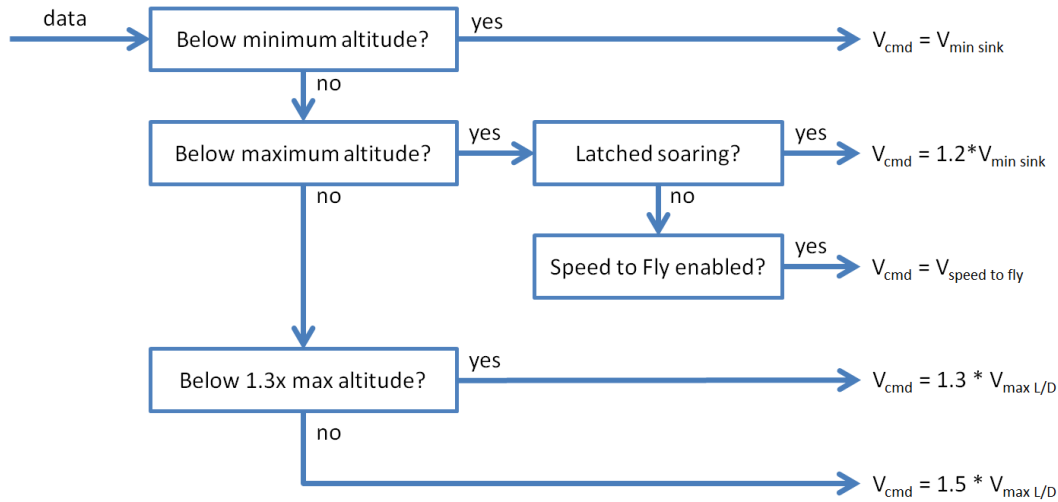

Fig. 15 — Airspeed command decision tree

The airspeed command decision tree essentially gives the aircraft a behavior of being conservative when below the minimum altitude (flying slowly to conserve altitude) and aggressive when above the maximum altitude (flying faster gets the vehicle out of any lift that may have taken it above the maximum altitude). During latched soaring, the airspeed command is always set to a margin above minimum sink speed to provide some maneuvering margin. Speed-to-fly is discussed in a separate section.

Speed to Fly

Using speed-to-fly gives the aircraft an ability to adjust its expenditure of stored energy altitude in an optimal fashion, depending on real-time measurements of current air conditions and expected air conditions. Compared to manually flown sailplanes, ALOFT can carry out the calculations in real time and continually adjust the airspeed command.

The calculation of speed-to-fly is based on a second-order polynomial approximation of the aircraft's sink polar [19]. If the polynomials are represented by

$$V_{sink} = aV^2 + bV + c \tag{94}$$

and a known headwind shifts the polynomials by

$$a_{wind} = a \tag{95}$$

$$b_{wind} = 2a * W_{hw} + b \tag{96}$$

$$c_{wind} = a * W_{hw}^2 + b * W_{hw} + c \tag{97}$$

then the speed-to-fly command is given by

$$V_{stf} = \sqrt{\frac{c_{wind} + w_t - w_{expect}}{a_{wind}}} + W_{hw} \tag{98}$$

where w_t is the current measurement of specific energy rate, w_{expect} is the speed-ring setting, and V_{stf} is the optimal speed-to-fly command for cross-country distance. The term w_{expect} is set using the speed-ring curve discussed in the Speed Ring Settings section.

Since upcoming air conditions are not known, w_{expect} is a guess. Opportunities for remote sensors that can directly measure upcoming air conditions (e.g., LIDAR [20]) are clearly visible here.

Speed Ring Settings

The MacCready speed ring is a sailplane-racing tool used to optimize the cross-country travel speed based on the aircraft's sink polar and the expected upcoming lift conditions, in combination with speed-to-fly. In cross-country racing, the selection of the MacCready setting can be critical. However, it does have applications for many more situations than simply racing.

The traditional speed ring is a movable bezel placed on the variometer instrument. The pilot dials the speed ring's zero-point to the expected climb rate in the next thermal. Then the variometer needle will point to a specific airspeed listed on the speed ring bezel [21]. For example, if the pilot sets the speed ring to zero, the variometer needle will point to commanded speeds faster than minimum sink when flying through sink. This airspeed gets the sailplane out of the sink area faster, resulting in a loss of less altitude than if the sailplane simply flew at maximum lift-to-drag ratio (L/D) or minimum sink.

For optimum performance, the speed-ring setting should be set to the average climb rate of the next encountered thermal [21]. That is, if the next thermal encountered will provide a 3 m/s climb rate, the speed ring should be set to 3 m/s and it will give an optimum cross-country speed-to-fly based on the current variometer reading.

Another incredibly useful output of the speed ring is that it optimally determines when to depart a thermal. In a given thermal, when the climb rate of the sailplane drops below the speed-ring setting, it is time to leave that thermal and transit to the next one. Correspondingly, during transit between thermals, if a thermal is encountered that is stronger than the current speed-ring setting, that thermal should be used.

The challenge of picking the correct speed-ring setting is that the next thermal conditions are unknown. "In optimizing any cross-country flight, the MacCready ring datum setting… is of prime importance" [21]. Lacking a remote sensor (such as a forward-looking LIDAR [20]), ALOFT used an empirically determined speed-ring-setting curve based on altitude. This provides a single curve that provides two behaviors:

- This curve defines an altitude-based threshold for latching and for unlatching.
- This curve also defines an altitude-based speed command.

For both behaviors, the net result is that the algorithm knows how to be conservative when at lower altitudes and aggressive when at higher altitudes. This set of behaviors made an enormous improvement in ALOFT's cross-country performance using the curve given in Fig. 16, which generally follows the inflection shown in Cochrane [22].
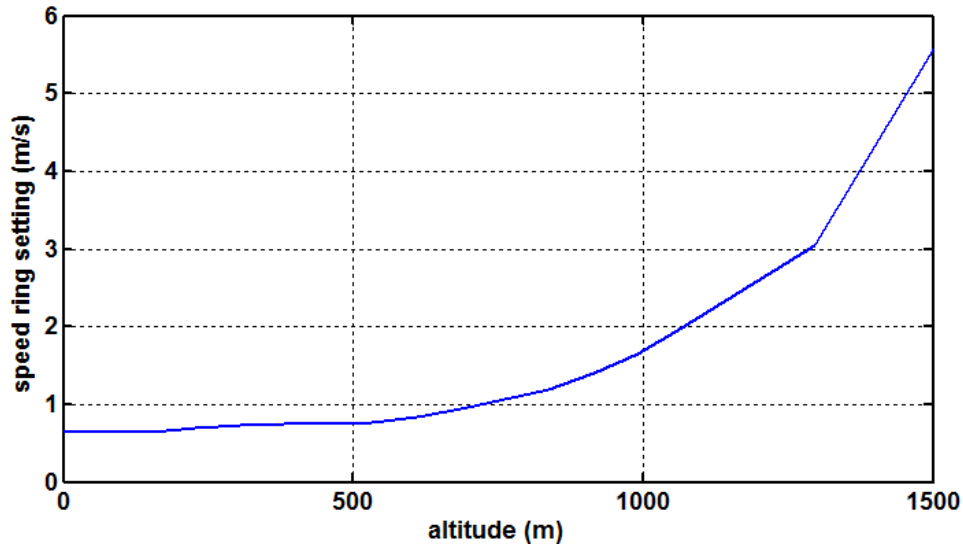

Fig. 16 — Speed ring setting based on altitude

The curve has three notable sections. First, the aircraft is fully conservative below 175 m, expecting only to find neutral air. Between 175 m and 600 m, the algorithm is progressively more aggressive with increasing altitude, but in effect wants to stay above 600 m. Above 600 m, the aggressiveness of the speed ring setting increases rapidly. The quality of lift above 600 m proved to be significantly better during flight-testing, so this became somewhat the operational floor above which the aggression was gradually increased. The limits of human vision tracking the SBXC occur around 1600 m, so the curve linearly climbs above 1300 m to infinity to ensure the aircraft pushes toward maximum speed before it has the chance to climb out of visibility. The curve presented here was developed empirically and does not represent an optimization.

Even when the task is maximum endurance rather than maximum cross-country speed, the speed ring helps get the vehicle through sink when traveling between thermals. The altitude curve shifts to be equally conservative at all altitudes, but the overall utility of the function is the same.

Future work should consider the concavity of the chosen speed ring curve. Cochrane's method [22] developed an optimal curve shape for maximizing racing points that has the opposite concavity. Perhaps a combination of the low-altitude saving behavior presented here and Cochrane's optimization could be used.

Future work could look at building the speed ring setting process based on a statistical measure of the encountered thermals. Knowing the inversion layer height, the progression of the strength of a thermal from ground to inversion layer height, and taking into account thermals that were encountered could perhaps tailor a "stock" curve online to change the vehicle's behavior more optimally over the course of a diurnal cycle. Also, future work might look at a remote sensor that measures the next thermal directly, such as a forward-looking LIDAR [20].

Orbit Radius

When in latched soaring mode, the orbit radius is selected using a simple altitude-scaling curve. Conventional soaring wisdom states that the orbit should be opened wider with an increase in altitude. ALOFT found that a minimum radius of 20 m was physically possible for the SBXC. Also, the Piccolo limits the orbit radius to 10 m increments. As a result, the orbit radius curve shown in Fig. 17 was used.
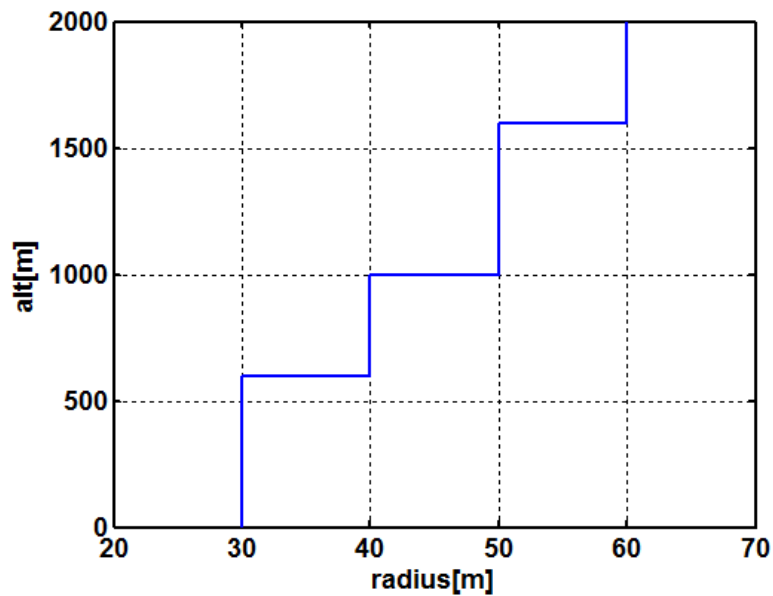

Fig. 17 — Orbit radius setting based on altitude

Future work might utilize the determined knowledge of the updraft radius as a way to set the orbit radius command. For a given sink polar and a given thermal cross section, an optimal bank angle can be determined [21], so there is certainly a more optimal way to select the orbit radius.

SEND TO PICCOLO

The Send to Piccolo functional block contains the low-level process of pushing commands to the autopilot. The only commands sent to the Piccolo are the following:

- Tracker command
  - Waypoint number that was previously being tracked (for when the aircraft is sent back into its previous flight plan after a soaring event).
  - The soaring orbit waypoint (assumes the waypoint was already uploaded).

- Waypoint command
  - Waypoint 98 is sent immediately upon latching and does not change from the start of the latched soaring event; this is for visually confirming the drift direction.
  - Waypoint 99 is sent each cycle through the control loop and is the current estimate of the thermal center, after being sent through the low-pass position filter. The waypoint command includes its position, orbit radius, orbit direction, and altitude.

- Airspeed command
  - If the airspeed command changes, it is sent to the autopilot.

Note that the engine kill command need not be sent to the autopilot. For a powered sailplane, the engine will naturally be reduced or turned off as the aircraft climbs above its altitude set-point. As such, all waypoints commands are always set to the minimum altitude so the vehicle turns the engine on when below the minimum and off when above the minimum.

CONCLUSIONS

This report documents the ALOFT autonomous soaring algorithm developed and flight tested on more than 100 flights of more than 70 hours total flight time. The algorithm's main functional blocks are to get a navigation solution from the autopilot, determine the center location of a nearby thermal, make some behavioral decisions for when to orbit, and then send the soaring commands to the autopilot. The modular nature of the control loop allows modifications on any of the sub-blocks without affecting the overall system.

Several areas of specific importance to the soaring algorithm have been identified, such as the thermal identification process using a combination of nonlinear least squares and a 3D to 2D transformation for nonlinear curve fitting, behavioral control using an altitude-based speed-ring curve, and the GoodLift soaring behavioral decision making process.

Areas of potential future improvements have also been identified, such as improvements in the updraft wind drift calculation, 4D nonlinear curve fitting of all four unknown parameters simultaneously, and remote sensing of upcoming thermal strengths for speed-to-fly calculation purposes.

The ALOFT algorithm was used on the world's first autonomous cross-country soaring sailplane, which has proven to outperform manually piloted remote-controlled aircraft in head-to-head competitions, such as the Montague Cross Country race 2010 [2], and which unofficially broke the goal-and-return cross-country record for a Fédération Aéronautique Internationale (FAI) glider aeromodel by flying for 60.4 mi round-trip.

It is hoped that the ALOFT autonomous soaring algorithm can be applied to a variety of UAVs in the future as a method of extending the endurance beyond what onboard energy stores allow. Perhaps the soaring techniques can even be applied to open-ocean environments and enable transoceanic flights by ever-smaller UAVs.

ACKNOWLEDGMENTS

REFERENCES

1. M. Allen, "Guidance and Control of an Autonomous Soaring UAV," NASA/TM-2007-214611, NASA Dryden Flight Research Center, Edwards, CA, Feb. 2007.

2. D.J. Edwards and L.M. Silberberg, "Autonomous Soaring: The Montague Cross-Country Challenge," *Journal of Aircraft* **47**(5), 1763–1769 (2010), doi: 10.2514/1.C000287.

3. J.W. Langelaan, "Long Distance/Duration Trajectory Optimization for Small UAVs," AIAA Guidance, Navigation, and Control Conference, 2007, doi: 10.2514/6.2007-6737.

4. N.T. Depenbusch and J.W. Langelaan, "Coordinated Mapping and Exploration for Autonomous Soaring," AIAA Infotech@ Aerospace Conference, 2011, doi: 10.2514/6.2011-1436.

5. K. Andersson, I. Kaminer, K.D. Jones, V. Dobrokhodov, and D.-J. Lee, "Cooperating UAVs Using Thermal Lift to Extend Endurance," AIAA Infotech@ Aerospace Conference, 2009, doi: 10.2514/6.2009-2043.

6. J. Glendening, "BLIPMAP Forecast Models," www.drjack.info/BLIP (accessed 16 Oct. 2014).

7. "XC Skies Soaring Forecast Maps & Tools," www.xcskies.com (accessed 16 Oct. 2014).

8. "Cloud Cap Technology," Hood River, Oregon, www.cloudcaptech.com (accessed 16 Oct. 2014).

9. S.C. Daugherty and J.W. Langelaan, "Improving Autonomous Soaring via Energy State Estimation and Extremum Seeking Control," AIAA Guidance, Navigation, and Control Conference, 2014, doi: 10.2514/6.2014-0260.

10. K. Andersson, I. Kaminer, V. Dobrokhodov, and V. Cichella, "Thermal Centering Control for Autonomous Soaring: Stability Analysis and Flight Test Results," *Journal of Guidance, Control, and Dynamics* **35**(3), May–June 2012, doi: 10.2514/1.51691.

11. E. Fonseka, "Modeling and Flying Thermal Tubes with a UAV," Semester Thesis Report, Autonomous Systems Lab, ETH Zürich, 2007.

12. Z. Hazen, "Design and Implementation of a Low Cost Thermal Soaring System for Uninhabited Aircraft," Thesis, University of Colorado, Boulder, 2007.

13. A.D. Kahn and D.J. Edwards, "Navigation, Guidance and Control for the CICADA Expendable Micro Air Vehicle," AIAA Guidance, Navigation, and Control Conference, 2012, doi: 10.2514/6.2012-4536.

14. M.W. Hazard, "Unscented Kalman Filtering for Real-Time Atmospheric Thermal Tracking," M.S. Thesis, North Carolina State University, Raleigh, NC, 2010.

15. D. Edwards, "Performance Testing of RNR's SBXC Using a Piccolo Autopilot," www.xcsoaring.com/tech.html (accessed 28 Aug. 2014).

16. A. Welch, I. Welch, and F. Irving, *The Complete Soaring Pilot's Handbook* (David McKay Company, Inc., New York, 1977).

17. "Coefficient of Determination," Wikipedia, http://en.wikipedia.org/wiki/Coefficient_of_determination (accessed 3 Sept. 2014).

18. E. Weisstein, "Least Squares Fitting," from Mathworld – A Wolfram Web Resource http://mathworld.wolfram.com/LeastSquaresFitting.html (accessed 3 Sept. 2014).

19. H. Reichmann, *Cross-Country Soaring* (Motorbuch Verlag, Stuttgart, Germany, 1975).

20. D.J. Edwards, "LIDAR for Remote Sensing of Updrafts for Improved Autonomous Soaring," NRL/MR/5710--12-9381, Naval Research Laboratory, Washington, DC, April 20, 2102.

21. F. Irving, *The Paths of Soaring Flight* (Imperial College Press, London, England, 2006).

22. J.H. Cochrane, "MacCready Theory with Uncertain Lift and Limited Altitude," *Technical Soaring* **23**, 88–96 (1999).

This page
intentionally
left blank

Appendix A

NONLINEAR LEAST SQUARES BY LINEAR ALGEBRA

Nonlinear regression can be accomplished using linear algebra, but it is not as robust nor as computationally fast as the series of sums. It is included here for completeness and clarity.

The data and fit parameters are expressed as

$$A\bar{\lambda} = \bar{b} \tag{A1}$$

where

$$A = \begin{bmatrix} 1 & D_1^2 \\ 1 & D_2^2 \\ \vdots & \vdots \\ 1 & D_n^2 \end{bmatrix} \tag{A2}$$

$$\bar{b} = \begin{bmatrix} ln(w_{t1}) \\ ln(w_{t2}) \\ \vdots \\ ln(w_{tn}) \end{bmatrix} \tag{A3}$$

$$\bar{\lambda} = \begin{bmatrix} ln(W) \\ R^{-2} \end{bmatrix} \tag{A4}$$

Note that this formulation includes a nonlinear transformation in order to use the linear least squares method to solve a nonlinear problem. This equation is solved for x using the pseudo-inverse:

$$\bar{\lambda} = (A^T A)^{-1} A^T \bar{b} \tag{A5}$$

The parameters W and R contained in λ are then back-solved through their respective transformations:

$$W = e^{\lambda_1} \tag{A6}$$

$$R = \sqrt{\frac{1}{\lambda_2}} \tag{A7}$$

Alternately, other least square regressions exist that are more robust to outliers, such as the Theil-Sen estimator, which could be used to improve performance of the estimator in linear space, before being transformed back to nonlinear space.

This page
intentionally
left blank

Appendix B

4D NONLINEAR THERMAL IDENTIFICATION

A 4D nonlinear regression has the capability to replace the entire evolutionary fitting method, instead solving for all four unknown parameters (W, R, x_0, y_0) at the same time. However, the method documented here has not proven to be as robust as the evolutionary method. It is included here for completeness, and for future modification that includes robustness improvements.

A nonlinear regression process can be performed to fit all four unknown parameters at the same time (W, R, x_0, y_0). It is seeded with the results of the 2D nonlinear regression.

The initial parameter guess is written as

$$\bar{\lambda} = \begin{bmatrix} W \\ R \\ x_0 \\ y_0 \end{bmatrix} \tag{B1}$$

Then, the following procedure is iterated for a maximum of 10 cycles. Starting with the Gaussian updraft model function:

$$w_t = W e^{-\left(\frac{D}{R}\right)^2} \tag{B2}$$

$$D = \sqrt{(x - x_0)^2 + (y - y_0)^2} \tag{B3}$$

The function is therefore

$$\bar{w}_t = W e^{\frac{-(\bar{x}-x_0)^2 - (\bar{y}-y_0)^2}{R^2}} \tag{B4}$$

This is evaluated with the currently guessed parameters, giving w_{guess}. The Jacobian matrix, A, is defined as

$$A = \begin{bmatrix} \frac{\partial w_t}{\partial W} & \frac{\partial w_t}{\partial R} & \frac{\partial w_t}{\partial x_0} & \frac{\partial w_t}{\partial y_0} \end{bmatrix}^T \tag{B5}$$

This matrix can be written as

$$A = diag\left(\left[e^{-\left(\frac{D}{R}\right)^2} \right]\right) \begin{bmatrix} 1 & \frac{2WD_1^2}{R^3} & \frac{2W(x_1-x_0)}{R^2} & \frac{2W(y_1-y_0)}{R^2} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \frac{2WD_n^2}{R^3} & \frac{2W(x_n-x_0)}{R^2} & \frac{2W(y_n-y_0)}{R^2} \end{bmatrix} \tag{B6}$$

where

$$\bar{D} = (\bar{x} - x_0)^2 + (\bar{y} - y_0)^2 \tag{B7}$$

The difference of the model to the current guess is then taken:

$$\bar{B} = w_t - w_{guess} \tag{B8}$$

Linear least squares is performed, the pseudo-inverse, to determine the fit parameters. Note that the inverse must be well conditioned.

$$\overline{\Delta\lambda} = (A^T A)^{-1} A^T \bar{B} \tag{B9}$$

The update is

$$\bar{\lambda} = \bar{\lambda} + \overline{\Delta\lambda} \tag{B10}$$

For the escape condition, the sum-squared error and the change in sum-squared error are computed:

$$SSE = \bar{B}^T \bar{B} \tag{B11}$$

The escape conditions from the iteration are when either of the following is satisfied:

$$|SSE| < 1 \tag{B12}$$

$$|\Delta SSE| < 0.01 \tag{B13}$$

Once out of the loop, the coefficient of determination, r^2, is then computed [B1]. This starts with the sum-squared error previously computed and the total sum of squares:

$$SST = \Sigma\left(\left(w_t - mean(\bar{w}_t)\right)^2\right) \tag{B14}$$

The coefficient of determination is

$$r^2 = 1 - \frac{SSE}{SST} \tag{B15}$$

The final estimated outputs are

$$W = \bar{\lambda}_1 \tag{B16}$$

$$R = \bar{\lambda}_2 \tag{B17}$$

$$x_0 = \bar{\lambda}_3 \tag{B18}$$

$$y_0 = \bar{\lambda}_4 \tag{B19}$$

Finally, the final estimated thermal position (x_0, y_0), strength (W), characteristic radius (R), and coefficient of determination (r^2) are given back to the calling function.

This 4D nonlinear regression is not well balanced, since W is typically on the order of 5, R is typically on the order of 150, and location varies with the distance from the home position up to 10,000.

A future research topic should include a method of weighting to get these parameters on the same magnitude so the adjustments occur evenly.

REFERENCE

B1. E. Weisstein, "Least Squares Fitting," from Mathworld – A Wolfram Web Resource http://mathworld.wolfram.com/LeastSquaresFitting.html (accessed 3 Sept. 2014).